

Detection de contours et restauration d'image par des algorithmes deterministes de relaxation. Mise en oeuvre sur la machine a connexions CM2

Josiane Zerubia, Florimond Ployette

► To cite this version:

Josiane Zerubia, Florimond Ployette. Detection de contours et restauration d'image par des algorithmes deterministes de relaxation. Mise en oeuvre sur la machine a connexions CM2. [Rapport de recherche] RR-1291, INRIA. 1990. inria-00075268

HAL Id: inria-00075268

<https://hal.inria.fr/inria-00075268>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE
IRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1291

Programme 6
Robotique, Image et Vision

DETECTION DE CONTOURS ET RESTAURATION D'IMAGE PAR DES ALGORITHMES DETERMINISTES DE RELAXATION. MISE EN ŒUVRE SUR LA MACHINE A CONNEXIONS CM2.

Josiane ZERUBIA
Florimond PLOYETTE

Octobre 1990



★ R R - 1 2 9 1 ★

Programme 6

**DETECTION DE CONTOURS ET
RESTAURATION D'IMAGE
PAR DES ALGORITHMES DETERMINISTES
DE RELAXATION.
MISE EN OEUVRE SUR LA MACHINE A
CONNEXIONS CM2.**

**EDGE DETECTION AND IMAGE
RESTORATION
USING TWO DETERMINISTIC
RELAXATION ALGORITHMS.
IMPLEMENTATION ON THE CONNECTION
MACHINE CM2.**

Josiane ZERUBIA

Florimond PLOYETTE

INRIA Sophia Antipolis
2004 Route des Lucioles
06560 Valbonne - FRANCE

Juillet 1990

abstract

Recently, a lot of algorithms minimizing a non-convex energy function have been proposed to solve low level vision problems. Different kinds of relaxation methods are available. The stochastic techniques, such as simulated annealing, asymptotically converge to the global minimum but require a high computational cost. Deterministic relaxation methods which are sub-optimal, give good results and are faster than the stochastic ones.

In this report, we focus on the parallel implementation of two deterministic algorithms for edge detection and image restoration: the graduated non-convexity (GNC) originally proposed by Blake & Zisserman and the mean field annealing (MFA) introduced by Geiger & Girosi and extended to anisotropic compound Gauss-Markov random fields by Zerubia & Chellappa. Both methods are based on a weak-membrane model and both algorithms are inherently serial: each step produces a pixel map which is taken as an input for the next step. For the GNC, we implement a checkerboard version of the successive over-relaxation (SOR) method to minimize the energy. For the MFA, we use an optimal step conjugate gradient descent.

Résumé

Récemment, de nombreux algorithmes de minimisation de fonctions non-convexes ont été proposés pour résoudre des problèmes de vision bas niveau. Il existe plusieurs méthodes de relaxation. Les techniques stochastiques, telles que le recuit simulé, convergent asymptotiquement vers le minimum global mais sont très coûteuses en temps de calcul. Les méthodes de relaxation déterministes, sont sous-optimales, mais donnent de bons résultats et sont plus rapides que les méthodes stochastiques.

Dans ce rapport, nous présentons la mise en oeuvre parallèle de deux algorithmes déterministes de détection de contours et de restauration d'image: le GNC (Graduated Non-Convexity) proposé par Blake & Zisserman et le recuit par champs moyens (MFA) introduit par Geiger & Girosi et étendu aux champs de Markov composés anisotropes par Zerubia & Chellappa. Ces deux méthodes sont fondées sur le modèle de la membrane à contraintes de continuité lâches et sont séquentiels: à chaque pas est produit une image qui est utilisée au pas suivant. Pour le GNC, nous avons utilisé une méthode de minimisation de l'énergie appelée "successive over-relaxation (SOR)" et plus précisément une variante parallèle deux couleurs. En ce qui concerne l'algorithme MFA, nous avons utilisé une méthode de descente de gradient conjugué à pas optimal.

Mots Clefs

Machine à Connexions, parallélisme de données, SIMD, Algorithmes déterministes de relaxation, détection de contours, restauration d'image.

Key Words

Connection Machine, Data parallelism, SIMD, Deterministic relaxation algorithms, Edge detection, Image restoration.

Remerciements

Les auteurs remercient David Ray, Harris Voorhees et James-Patrick Massar de Thinking Machine Corporation pour leurs conseils ainsi que la SAGEM pour la fourniture des images aériennes présentées dans les annexes 4 et 5 (propriété SAGEM). Le premier auteur souhaite également remercier Rama Chellappa, Directeur du "Signal and Image Processing Institute" de USC (LA, USA) pour son accueil. Les algorithmes séquentiels dont les résultats sont présentés dans ce rapport ont été développés lorsque le premier auteur était en année post-doctorale à USC.

Acknowledgements

The authors would like to thank David Ray, Harris Voorhees and James-Patrick Massar of Thinking Machine Corporation for many useful discussions as well as SAGEM for providing the aerial images presented in annex 4 and 5 (property of SAGEM). The first author also thanks Rama Chellappa, Director of the Signal and Image Processing Institute of USC (LA, USA) for having received her in his laboratory. The sequential algorithms whose results are presented in this report have been implemented by the first author during her post-doctoral year at USC.

<i>TABLE DES MATIÈRES</i>	4
---------------------------	---

Table des matières

1 Introduction	5
2 Présentation des deux méthodes déterministes de relaxation	5
2.1 Rappel sur les champs de Markov et les processus de ligne	5
2.2 GNC	7
2.3 Recuit par champs moyens	10
3 La Machine à Connexions	12
4 Mise en oeuvre des algorithmes de relaxation sur CM2	14
4.1 Primitives de traitement d'image sur la CM2	14
4.1.1 Entrée-Sortie d'image	14
4.1.2 Opérations arithmétiques et logiques sur les variables parallèles . . .	14
4.1.3 Communications inter-processeurs	14
4.1.4 Opérations globales de réduction	15
4.2 Mise en oeuvre des algorithmes GNC et MFA	15
5 Performance des différents algorithmes	16
5.1 Choix des paramètres	17
5.2 Comparaison des temps de calcul	18
5.3 Comparaison de la qualité des résultats	20
6 Conclusion	22
7 Bibliographie	22
A Annexe 1	25
B Annexe 2	27
C Annexe 3	32
D Annexe 4	39
E Annexe 5	46

1 Introduction

L'utilisation des champs de Markov [13] en vision par ordinateur s'est développée depuis quelques années. Les principales applications de ce type de modélisation sont la détection de contours [3], [7], [23], [28], la restauration d'image [3], [8], [18], [23], [28], la reconstruction de surface [3], [7], la stéréoscopie [27], la synthèse et la classification de texture [5], [9], [21].

Le principal avantage des champs de Markov est de fournir un modèle mathématique simple permettant de prendre en compte les interactions locales au niveau du pixel. Un inconvénient majeur est le coût de calcul relatif à tout algorithme obtenu à partir d'une telle modélisation. C'est pourquoi le développement sur des machines parallèles d'algorithmes basés sur des champs de Markov est intéressant.

Dans ce rapport, nous présentons les résultats obtenus en détection de contours et restauration d'image grâce à deux méthodes déterministes de relaxation: le GNC [3], [23] et le recuit par champs moyens [7], [28]. Ces algorithmes itératifs ont été mis en oeuvre sur la machine à connexions CM2 de l'INRIA et comparés à un algorithme classique de détection de contours: le filtre de Canny mis en oeuvre sur cette machine par Thinking Machine Corporation (TMC).

2 Présentation des deux méthodes déterministes de relaxation

2.1 Rappel sur les champs de Markov et les processus de ligne

Soit une image $\{u(s)\}$ définie sur un réseau fini Ω de sites $s : \Omega = \{s = (i, j); 1 \leq i, j \leq M\}$.

Considérons une famille arbitraire N de voisinages locaux de $\Omega : N = \{N(s), s \in \Omega\}$.

A chaque famille de voisinages est associée une classe de distribution de probabilités, appelée champ de Markov et caractérisée par la propriété suivante:

$$P(u(s)|u(r), r \in \Omega - s) = P(u(s)|u(r), r \in N(s) - s) \quad (1)$$

Ce qui signifie que la connaissance d'un voisinage local du pixel s est suffisante pour calculer la probabilité en chaque pixel s .

2 PRÉSENTATION DES DEUX MÉTHODES DÉTERMINISTES DE RELAXATION 6

Une autre propriété importante est le théorème de Hammersley-Clifford [1] qui montre qu'un champ aléatoire défini sur un réseau est un champ de Markov si et seulement si sa distribution de probabilité est une distribution de Gibbs définie par:

$$P(u) = \frac{\exp - \frac{E(u)}{T}}{Z} \quad (2)$$

où $E(u)$ est une fonction énergie définie par:

$$E(u) = \sum_{c \in C} V_c(u) \quad (3)$$

où C est l'ensemble de cliques correspondant au voisinage choisi (i.e. \bullet et $\bullet\bullet$ pour un voisinage du premier ordre).

$V_c(u)$ est une fonction potentiel permettant d'exprimer les contraintes et T est la température. Z est appelé la fonction de répartition. C'est une constante définie par:

$$Z = \sum_{\text{toutes les configurations}} \exp - \frac{E(u)}{T} \quad (4)$$

Nous supposons que l'image observée est décrite par un champ d (par exemple: $d = u + n$ où n est un bruit blanc Gaussien). En utilisant le théorème de Bayes, il vient:

$$P(u|d) = \frac{P(d|u)P(u)}{P(d)} = \frac{\exp - \frac{E(u|d)}{T}}{Z} \quad (5)$$

or $P(d)$ est constant et $P(d | u)$ est connu puisque un modèle a priori a été choisi (i.e. bruit additif gaussien).

L'estimée \hat{u} de u peut être obtenue par la maximisation de la probabilité a posteriori (MAP) ce qui revient à minimiser l'énergie $E(u | d)$.

Il s'avère intéressant de définir un champ booléen dual au processus intensité, appelé processus de ligne. Ce champ, introduit pour la première fois par Geman & Geman [8], représente explicitement la présence ou l'absence de discontinuité et vient ainsi rompre l'hypothèse de lissage faite par les méthodes classiques de régularisation [17].

2 PRÉSENTATION DES DEUX MÉTHODES DÉTERMINISTES DE RELAXATION

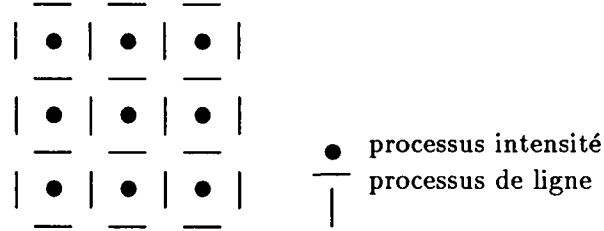


Figure 1: Superposition des 2 réseaux

Dans le modèle considéré, nous avons deux processus de ligne (horizontal et vertical, cf. Figure 1). L'énergie peut alors s'exprimer par:

$$E = D + S + P \quad (6)$$

avec $D = \sum_{i,j} (u_{i,j} - d_{i,j})^2$ qui assure une bonne adéquation de la solution trouvée aux données.

$$S = \sum_{i,j} \lambda^2 ((u_{i,j} - u_{i-1,j})^2 (1 - l_{i,j}) + (u_{i,j} - u_{i,j+1})^2 (1 - m_{i,j}))$$

qui représente la contrainte de lissage classique ($\lambda^2 \star \text{gradient}^2$) mais qui tient également compte de l'interaction entre le champ intensité et le champ processus de ligne.

$P = \sum_{i,j} \alpha(l_{i,j} + m_{i,j})$ qui est le coût à payer pour l'introduction d'un contour.

Le modèle décrit par l'équation (6), est celui d'une membrane à contraintes de continuité lâches [3], [7], [8].

2.2 GNC

Pour minimiser l'énergie décrite par l'équation (6) qui est une fonction non-convexe, Blake et Zisserman [3] ont proposé un algorithme déterministe de relaxation, appelé GNC

(“Graduated Non Convexity”).

Les techniques classiques de minimisation consistent soit en l’application d’une méthode déterministe de type descente suivant la plus grande pente (auquel cas, le système peut être bloqué dans un minimum local quelconque), soit en l’utilisation d’une technique stochastique de recuit simulé [8], [12], [14] (qui assure la convergence asymptotique vers le minimum global mais qui demande un coût de calcul très élevé).

La méthode du GNC assure de trouver le minimum global dans des cas particuliers (par exemple si les discontinuités sont assez éloignées les unes des autres) et donne de bonnes solutions dans les autres cas (cf. Chapitre 7 de [3] pour plus de détails). Nous en rappelons brièvement le principe ci-dessous.

L’équation (6) peut se mettre sous la forme:

$$\begin{aligned} E = D + \sum_{i,j} h_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}, l_{i,j}) \\ + \sum_{i,j} h_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}, m_{i,j}) \end{aligned} \quad (7)$$

avec

$$h_{\alpha,\lambda}(t, l) = \lambda^2(t)^2(1 - l) + \alpha l \quad (8)$$

La minimisation de E doit être faite par rapport à $u_{i,j}$, $l_{i,j}$ et $m_{i,j}$. Cependant, le terme D ne contient pas de processus de ligne. La minimisation par rapport à $l_{i,j}$ et $m_{i,j}$ peut être effectuée avant celle par rapport à $u_{i,j}$:

$$\begin{aligned} \min_{(u_{i,j})} E = \min_{(u_{i,j})} (D + \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) \\ + \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i,j+1})) \end{aligned} \quad (9)$$

avec

$$g_{\alpha,\lambda}(t) = \min_{l \in (0,1)} h_{\alpha,\lambda}(t, l) \quad (10)$$

2 PRÉSENTATION DES DEUX MÉTHODES DÉTERMINISTES DE RELAXATION

L'idée de base du GNC est la suivante: la première étape est la construction d'une approximation convexe E^* de l'énergie E , la minimisation de E^* donne donc un minimum global. Puis une séquence d'énergies $E^{(p)}$ est construite telle que $E^1 = E^*$ et $E^0 = E$. Le GNC optimise toute la séquence d'énergies $E^{(p)}$ (par exemple, $p=1, 1/2, 1/4, 1/8...$ pour une image) en utilisant comme conditions initiales:

- Les données pour la minimisation de E^*
- La solution optimale trouvée lors de la minimisation de la fonction énergie précédente dans les autres cas.

Blake et Zisserman ont montré dans [3] que la fonction $g_{\alpha,\lambda}^{(p)}$ est la suivante:

$$g_{\alpha,\lambda}^{(p)}(t) = \begin{cases} \lambda^2(t)^2 & |t| < q \\ \alpha - c(|t| - r)^2/2 & q \leq |t| < r \\ \alpha & |t| \geq r \end{cases}$$

avec

$$c = \frac{1}{4p}, r^2 = \alpha\left(\frac{2}{c} + \frac{1}{\lambda^2}\right), q = \frac{\alpha}{\lambda^2 r} \quad (11)$$

où $t = (u_{i,j} - u_{i-1,j})$ ou $t = (u_{i,j} - u_{i,j+1})$ suivant la direction considérée.

Une fois la minimisation par rapport au champ intensité $u_{i,j}$ effectuée sur toute la série d'énergies $E^{(p)}$, les processus de ligne (donc les contours) sont obtenus de la façon suivante:

$$l_{i,j} = \begin{cases} 1 & \text{si } |u_{i,j} - u_{i-1,j}| > r \\ 0 & \text{si } |u_{i,j} - u_{i-1,j}| < q \\ \text{ambigu} & \text{sinon} \end{cases}$$

Nous avons pris $l_{i,j} = 0$ pour le cas ambigu lors de la mise en oeuvre de l'algorithme. Pour obtenir $m_{i,j}$, il suffit de faire le même test en remplaçant $u_{i-1,j}$ par $u_{i,j+1}$.

Une extension de cette méthode au cas d'un modèle anisotrope de champs de Markov (caractérisé par θ_x, θ_y et une variance ν) a été réalisé par Simchony et al. [23]. Elle n'a pas été mise en oeuvre sur la machine à connexions car le coût de calcul a été jugé trop important par rapport au gain de qualité obtenu pour la détection de contours et la restauration d'image (des tests ayant été effectués en séquentiel à USC).

2.3 Recuit par champs moyens

Une autre méthode déterministe de relaxation basée sur un recuit par champs moyens ("Mean Field Annealing") a été introduite par Geiger et Girosi [6] pour minimiser l'équation (6).

La technique d'approximation par champs moyens est souvent utilisée en physique statistique [19], [20]. Elle a également été employée récemment pour résoudre des problèmes combinatoires comme par exemple le partitionnement de graphe [2], [10] ou le stockage d'exemples ("pattern storage") [4] en utilisant des réseaux de neurones. Dans tous les cas, cette technique s'est avérée plus rapide par le recuit simulé (cf. [2] par exemple).

Par définition, le champ moyen $\bar{u}_{i,j}$ est donné par:

$$\bar{u}_{i,j} = \frac{\sum_{toutes\ les\ config.} u_{i,j} \exp - \frac{E(u,l,m)}{T}}{Z} \quad (12)$$

$E(u, l, m)$ étant donné par l'équation (6). Les champs moyens $\bar{l}_{i,j}$ et $\bar{m}_{i,j}$ sont définis de façon analogue.

Après quelques calculs (cf. [6] pour plus de détails), on peut montrer que:

$$\bar{u}_{i,j} = d_{i,j} + \frac{T}{2} \frac{\partial \ln Z}{\partial d_{i,j}} \quad (13)$$

et

$$\bar{l}_{i,j} = 1 - T \frac{\partial \ln Z}{\partial G_{i,j}^l} \quad (14)$$

où

$$G_{i,j}^l = \alpha - \lambda^2 (u_{i,j} - u_{i-1,j})^2 \quad (15)$$

une équation similaire étant définie pour $\bar{m}_{i,j}$.

Si nous voulons effectivement obtenir ces champs moyens, nous devons faire des approximations afin de calculer la fonction de répartition Z de façon explicite.

2 PRÉSENTATION DES DEUX MÉTHODES DÉTERMINISTES DE RELAXATION 11

Considérons tout d'abord le calcul de $\bar{u}_{i,j}$, la fonction de répartition Z est définie par:

$$Z = \sum_{\text{toutes les config. } u, l, m} \exp - \frac{E(u, l, m)}{T} \quad (16)$$

La première étape est d'éliminer les processus de ligne dans la fonction énergie E telle que:

$$E(u, l, m) = D(u) + E^{eff}(u) \quad (17)$$

Ceci peut être calculé en considérant les processus de ligne l et m comme des verres de spin dans un champ extérieur (respectivement $G_{i,j}^l$ et $G_{i,j}^m$) sans interaction entre voisins (ce qui est vrai car aucune interaction entre les processus de ligne n'est supposée) - cf [6] et [7] pour plus de détails.

La seconde étape est d'approximer Z :

$$Z = \sum_{\text{toutes les config. } u} \exp - \frac{D(u) + E^{eff}(u)}{T} \quad (18)$$

$$Z \approx \max_{(u)} \left(\exp - \frac{\sum_{i,j} (u_{i,j} - d_{i,j})^2 + E_{i,j}^{eff}(u)}{T} \right) \quad (19)$$

$$Z \approx \exp - \frac{\sum_{i,j} (\bar{u}_{i,j} - d_{i,j})^2 + E_{i,j}^{eff}(\bar{u})}{T} \quad (20)$$

En pratique, cela revient à dire qu'au lieu de raisonner sur l'influence stochastique des champs des pixels voisins sur le champ du pixel (i, j) , il suffit de raisonner sur l'influence exercée par les valeurs moyennes des champs des pixels voisins. Cela revient à "figer" l'environnement du pixel (i, j) . Cette approximation n'est valable bien sûr que lorsque T tend vers 0.

Pour obtenir $\bar{u}_{i,j}$, il suffit donc de minimiser l'énergie en faisant :

$$\frac{\partial (D(u) + E^{eff}(u))}{\partial u_{i,j}} = 0 \quad (21)$$

ce qui donne:

$$\begin{aligned}
\bar{u}_{i,j} = & d_{i,j} - \lambda^2(\bar{u}_{i,j} - \bar{u}_{i,j+1})(1 - \bar{m}_{i,j}) \\
& + \lambda^2(\bar{u}_{i,j-1} - \bar{u}_{i,j})(1 - \bar{m}_{i,j-1}) \\
& - \lambda^2(\bar{u}_{i,j} - \bar{u}_{i-1,j})(1 - \bar{l}_{i,j}) \\
& + \lambda^2(\bar{u}_{i+1,j} - \bar{u}_{i,j})(1 - \bar{l}_{i+1,j})
\end{aligned} \tag{22}$$

En partant de l'équation (14) et en utilisant l'approximation par champs moyens décrite ci-dessus, il vient après quelques calculs:

$$\bar{l}_{i,j} = \sigma_\beta(\lambda^2(\bar{u}_{i,j} - \bar{u}_{i-1,j})^2 - \alpha) \tag{23}$$

$$\bar{m}_{i,j} = \sigma_\beta(\lambda^2(\bar{u}_{i,j} - \bar{u}_{i,j+1})^2 - \alpha) \tag{24}$$

avec $\beta = \frac{1}{T}$ et $\sigma_\beta(x) = \frac{1}{1+\exp-\beta x}$

Il faut remarquer que, contrairement à la méthode du GNC, les processus de lignes ne sont plus des champs booléens mais des champs continus à valeur dans $[0,1]$. Ce n'est que lorsque la température T tend vers 0 (i.e. $\beta \rightarrow \infty$) que la sigmoïde devient un échelon rendant ainsi les processus de ligne booléens.

Une extension de cette méthode au cas d'un modèle anisotrope de champs de Markov a été réalisée par Zerubia et Chellappa [28], [29]. Elle n'a pas été mise en oeuvre sur la machine à connexions pour les mêmes raisons que celles invoquées au chapitre 2.2 pour le GNC.

3 La Machine à Connexions

Dans ce paragraphe, nous donnons une description succincte de cette machine dont on trouvera une documentation plus complète en [11], [24], [25]. La machine à connexions est une machine SIMD ("Single Instruction Multiple Data") ayant de 8K à 64K processeurs. Chaque processeur est un processeur 1 bit disposant de 32K octets de mémoire locale et une horloge à 8 MHz. Les macro-instructions provenant du calculateur frontal sont traitées par un micro-contrôleur qui diffuse des nano-instructions à tous les processeurs.

L'organisation physique de l'architecture est la suivante (cf Figure 2):

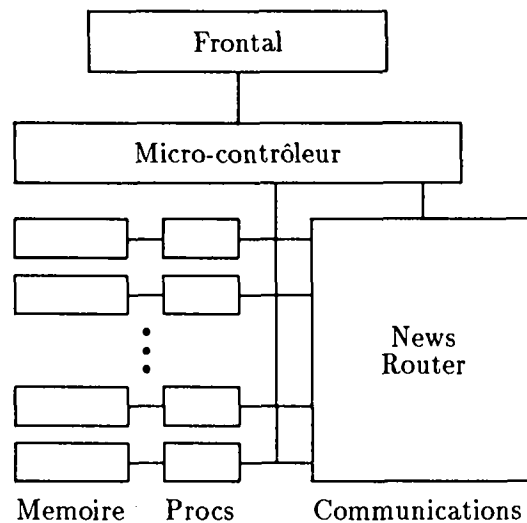


Figure 2: Architecture de la CM-2

- Une puce regroupe 16 processeurs.
- Une section regroupe 2 puces de 16 processeurs, la mémoire locale de ces processeurs et une autre puce contenant l'unité de calcul flottant.
- Les puces processeurs (contenant chacune 16 processeurs) sont interconnectées en un hypercube à 12 dimensions (il y a de 1K à 4K puces processeurs). Chaque processeur a une adresse codée sur 16 bits et deux processeurs sont directement reliés si leur adresse diffère de 1 bit.

Pour chaque application, l'utilisateur peut dynamiquement définir une géométrie particulière pour l'ensemble des processeurs qu'il utilise. Cette géométrie décrit l'organisation logique de ces processeurs: le nombre de dimensions de la structure et le nombre de processeurs par dimension. Il peut, en outre, s'affranchir de la limite imposée par le nombre de processeurs physiques grâce au mécanisme des processeurs virtuels. La machine à connexions est donc une machine à parallélisme de données. Une même opération est effectuée en parallèle sur une grande masse de données. Ce type de parallélisme est bien adapté au traitement d'image bas niveau puisque les données (l'image) sont importantes et que les traitements sont locaux et identiques pour chaque pixel [15].

4 Mise en oeuvre des algorithmes de relaxation sur CM2

Le GNC (“Graduated Non Convexity”) et le recuit par champs moyens (“Mean Field Annealing”) ont été implémentés sur la machine à connexions. Cette mise en oeuvre a été réalisée en *lisp [24] qui est une extension de Common Lisp pour permettre la programmation parallèle sur la machine à connexions. Dans ce paragraphe, nous présentons d’abord les primitives que nous employons pour le traitement d’image en général, puis nous décrivons la mise en oeuvre des deux algorithmes.

4.1 Primitives de traitement d’image sur la CM2

4.1.1 Entrée-Sortie d’image

Pour effectuer un traitement sur une image il convient d’abord de la charger dans la mémoire de la machine. Dans un premier temps, il faut configurer la machine suivant une géométrie adaptée aux dimensions de l’image. En général, la géométrie retenue est une grille à deux dimensions dont la taille est égale à la puissance de deux immédiatement supérieure au nombre de pixels. On obtient donc un nombre de processeurs virtuels supérieur ou égal au nombre de pixels de l’image. Le fichier image qui a été lu sur le calculateur frontal est un tableau dont chaque élément(pixel) est chargé sur un processeur de la géométrie. Ce transfert utilise un bus global qui relie le frontal à la mémoire des processeurs. La représentation de l’image dans la machine est alors une variable parallèle (pvar en *lisp) dont chaque élément est un pixel. L’opération d’écriture d’une image depuis sa représentation dans la CM dans un fichier image est l’opération inverse.

Une méthode plus rapide consiste à stocker l’image sous forme d’un fichier parallèle dans le système de fichiers CM sur le “Data Vault” [24]; la lecture et l’écriture de l’image sont alors plus rapides.

4.1.2 Opérations arithmétiques et logiques sur les variables parallèles

Toutes les opérations arithmétiques et logiques classiques sont disponibles pour les variables parallèles (pvar). Ces opérations prennent en général deux variables parallèles et délivrent en résultat une troisième variable parallèle. Ce sont les opérations de base de la CM qui sont effectuées en parallèle sur tous les processeurs.

4.1.3 Communications inter-processeurs

Il existe plusieurs types de communications entre les processeurs. La plus générale étant l’envoi en parallèle d’une valeur de chaque processeur vers n’importe quel autre. Par

exemple l'expression *lisp suivante:

```
(*pset pvar1 pvar2 dest-address-pvar)
```

où pvar1, pvar2 sont des variables parallèles et dest-address-pvar est également une variable parallèle contenant des adresses de processeurs. L'effet de cette instruction est d'envoyer chaque valeur de pvar1 dans l'élément de la variable parallèle pvar2 dont l'adresse processeur est donnée dans dest-address-pvar. L'ensemble des valeurs de pvar1 subit donc une permutation si toutes les adresses destination sont distinctes. Si certaines adresses destination sont identiques il y a collision et il est possible d'associer des opérations courantes (+, max, min, ou etc ...) de manière à combiner les différentes valeurs destinées à un processeur en une seule valeur. Nous utilisons ce type de communication générale pour dilater ou réduire une image.

Le type de communication que nous utilisons le plus en traitement d'image bas niveau est la communication locale et relative à la géométrie choisie. Cette communication appelée NEWS (pour "North, East, West, South") est très rapide car, dans la plupart des cas, le transfert d'information se fait entre des processeurs situés sur la même puce et par un circuit spécialisé différent du circuit de routage général.

4.1.4 Opérations globales de réduction

Ces opérations consistent à effectuer une opération élémentaire (+, -, max, min, ou, et, ...) successivement sur tous les éléments d'une variable parallèle et de délivrer une seule valeur qui est transmise au frontal par le bus global. Ces opérations sont très efficaces car leur mise en oeuvre est parallélisée.

4.2 Mise en oeuvre des algorithmes GNC et MFA

D'un point de vue mise en oeuvre, ces deux algorithmes sont assez semblables puisqu'il s'agit de deux algorithmes de relaxation itératifs. A l'issue de chaque itération, une nouvelle image restaurée est calculée et est utilisée à l'itération suivante. La valeur de la fonction d'énergie est calculée à chaque itération et sert dans le test d'arrêt de la boucle. Le calcul de cette valeur nécessite une opération globale de réduction sur toute l'image (une somme globale).

Dans l'algorithme GNC, le calcul du gradient se fait en deux passes successives. On considère l'image comme un échiquier, le calcul du gradient est d'abord effectué sur les

cases blanches de cet échiquier, puis, à la seconde passe, sur les cases noires. Ces deux passes sont séquentielles car, à la seconde passe on utilise le résultat du gradient calculé à la première passe. En effet, chaque processeur communique avec ses quatre voisins cardinaux. La minimisation de l'énergie se fait par une méthode de type SOR ("Successive Over-Relaxation") cf Figure 7.13 dans [3].

Dans l'algorithme MFA on utilise une méthode de descente optimale par gradient conjugué [16], [22] qui nécessite un nombre plus élevé de calcul de l'énergie.

Le tableau 1 montre le nombre d'instructions de chaque type pour chacune des itérations. Toutes les instructions sont flottantes sur 32 bits. Dans ce tableau ne figurent pas les instructions de gestion de la pile et de contrôle. Il est intéressant de noter que le rapport entre le nombre d'instructions de communication et le nombre d'instructions flottantes est très faible et que ces communications sont locales.

	Op.Arith.et Compar.	NEWS	Op.Global
GNC	117	10	1
MFA	123	24	4

Table 1 : Nombre d'instructions CM par type pour chaque itération.

5 Performance des différents algorithmes

Nous avons testé ces algorithmes sur 6 images que nous avons sélectionnées pour leur complémentarité :

- **image aérienne (128,128) bruitée artificiellement par un générateur de bruit blanc Gaussien (rapport signal/bruit = 5 dB)** : l'intérêt de cette image est le respect des hypothèses faites pour les développements mathématiques du Chapitre 2. Elle constitue un "cas d'école" pour la restauration d'image et la détection de contours en milieu bruité. Les résultats obtenus sont présentés dans l'Annexe no 1.
- **image d'intérieur (512,512) sans bruit** : cette image est également un "cas d'école". Il est bien connu que les algorithmes de type Canny donnent de bons résultats pour la détection de contours sur ce type d'image (la restauration étant ici sans objet). Leur seul défaut réside dans la non-fermeture des contours. Les images de contours obtenues avec les différentes méthodes sont présentées dans l'Annexe no 2.

- **image spot (512,512) sans bruit** : cette image satellite contient des routes, des rivières, des parcelles ainsi que des habitations qui créent localement une texture. Les résultats obtenus sont présentés en Annexe no 3.
- **image ign (1024,1024) sans bruit** : cette image est du même type que la précédente, nous l'avons choisie pour évaluer les temps de calcul des différents algorithmes sur la CM2 lorsque le nombre de pixels à traiter est élevé. Les résultats sont présentés dans la Table 5 et la Figure 3.
- **image aérienne floue (512,512)** : l'intérêt de cette image est de présenter un flou important dont on ne connaît pas la Densité Spectrale de Puissance (DSP). Nous avons décidé d'utiliser le même modèle que pour les autres images. Nous aurions pu développer un modèle plus proche de la réalité, mais il aurait alors fallu estimer les paramètres de la DSP du flou par une méthode du maximum de vraisemblance par exemple et modifier en conséquence les algorithmes proposés ce qui aurait demandé trop de temps. Les résultats obtenus sont présentés dans l'Annexe no 4.
- **image aérienne tramée (512,512)** : comme précédemment nous n'avons pas défini de modèle particulier pour cette image. Nous présentons les résultats en Annexe no 5.

Dans un premier temps, nous présentons le choix des paramètres effectué pour chaque algorithme. Puis, nous nous intéressons au temps de calcul des différents algorithmes ; nous comparons les algorithmes entre eux, puis nous comparons l'approche parallèle avec l'approche séquentielle [30]. Enfin, nous comparons ces algorithmes du point de vue de la qualité des résultats.

5.1 Choix des paramètres

Pour la méthode du GNC, les deux paramètres à choisir sont le facteur de lissage λ qui est également un facteur d'échelle (cf [3]) et le seuil $h = \sqrt{\frac{2\sigma}{\lambda}}$ qui fixe la limite de détection des contours. Le test d'arrêt de l'algorithme est basé sur un seuillage de la variation d'énergie ΔE calculée après chaque itération. La Table 2 donne les valeurs des paramètres utilisés pour chaque image (A1 = images contenues dans l'Annexe no 1,...etc).

Pour la méthode du recuit par champs moyens (MFA), le choix de la loi de variation de température est très important. En effet, lorsque les images sont peu contrastées ou fortement bruitées, une variation trop rapide de la température donne des images restaurées de mauvaise qualité ainsi qu'une détection de contours moyenne alors que le test de convergence de l'algorithme (basé sur la variation d'énergie ΔE) est correct. Après avoir testé

Image	A1	A2	A3	A4	A5
λ	4	2	4	3	4
h	16	6	14	14	16
ΔE	200	150	150	150	150

Table 2 : Paramètres pour le GNC

différentes lois de variation de température proposées dans la littérature, nous avons choisi une loi empirique qui donne de bons résultats sur toutes les images testées:

- $\beta = 0.0002$ à l'initialisation.
- $\beta = \beta * 4$ tant que $\beta \leq 1$.

Les paramètres à fixer sont le facteur de lissage λ et le coût α à payer pour créer une discontinuité (ce qui revient à choisir h). Pour toutes les images présentées en annexe, nous avons pris $\lambda^2 = 6$, $\alpha = 180$ et $\Delta E = 100$.

Enfin, pour le filtre de Canny proposé par TMC, les deux seuils sont calculés automatiquement (un seul paramètre est à fixer, nous avons choisi $\sigma = 1.5$ pour toutes les images).

5.2 Comparaison des temps de calcul

Nous comparons trois algorithmes de détection de contours mis en oeuvre sur la machine à connexions:

- Le détecteur de Canny développé par TMC.
- L'algorithme GNC.
- L'algorithme de recuit par champs moyens (MFA).

Pour chacun de ces algorithmes, nous donnons:

- le VPR, c'est à dire le nombre de processeurs virtuels par processeurs réels ou le nombre de pixels par processeurs.
- Le temps CM, c'est le temps d'exécution des instructions CM.
- Le temps total.
- Le nombre d'itérations de l'algorithme pour le GNC et le MFA.

- Le temps CM par itération.

Les tables 3, 4 et 5 montrent ces résultats sur trois images de taille différente: une image 128x128, une image 512x512 et une image 1024x1024.

	VPR	Tps.CM	Tps.Total	Nb.Iter	Tps.CM par It
Canny	2	0.09	0.12		
GNC	2	21	64.5	567	0.03
MFA	2	10.2	19.13	174	0.05

Table 3 : Image aérienne bruitée 128x128

	VPR	Tps.CM	Tps.Total	Nb.Iter	Tps.CM par It
Canny	32	0.66	0.69		
GNC	32	91.5	97.9	187	0.49
MFA	32	166.8	203	278	0.60

Table 4 : Image d'intérieur non bruitée 512x512

	VPR	Tps.CM	Tps.Total	Nb.Iter	Tps.CM par It
Canny	128	4.12	4.37		
GNC	128	302.13	320.8	156	1.93
MFA	128	947.64	1048.7	411	2.3

Table 5 : Image ign non bruitée 1024x1024

	Fortran-Sun4		*Lisp-CM	
	Tps.Total	Tps.par it.	Tps.Total	Tps.par it.
GNC	15mn.	1.69s	64.5s	0.11s
MFA	25mn.	7.97s	19.13s	0.109s

Table 6 : Image aérienne 128x128 Fortran-Sun4 et *Lisp-CM

	Fortran-Sun4		*Lisp-CM	
	Tps.Total	Tps.par it.	Tps.Total	Tps.par it.
GNC	50mn	19.86s	97.9s	0.52s
MFA	8H40mn	1mn52s	203s	0.73s

Table 7 : Image d'intérieur 512x512 Fortran-Sun4 et *Lisp-CM

La première remarque est de constater la différence de rapidité entre l'algorithme de Canny et les deux autres. Ces algorithmes sont en effet de nature très différente puisque

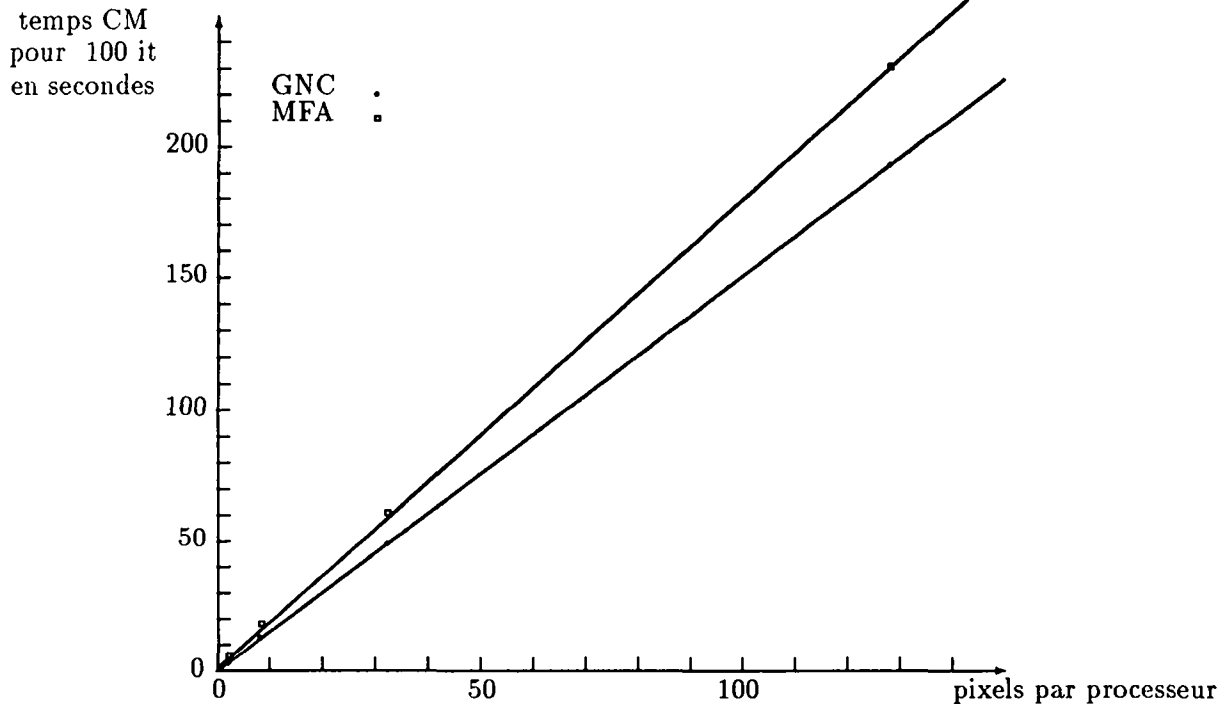


Figure 3: Comparaison des temps de calcul des algorithmes GNC et MFA

l'algorithme de Canny n'est pas un algorithme itératif contrairement aux autres. Cette pénalité en temps de calcul est compensée par une qualité supérieure, surtout dans le cas bruité, comme nous le discutons au paragraphe suivant.

La seconde remarque est de noter que les deux algorithmes de relaxation GNC et MFA sont sensiblement équivalents au point de vue temps de calcul, avec toutefois un petit avantage pour le GNC qui comporte moins d'instructions par itération. La Figure 3 résume bien ce comportement. On constate en effet que, pour chacun des deux algorithmes, le temps par itération est proportionnel au nombre de pixels par processeur. Toutefois, lorsque le nombre de pixels croît, le rapport temps de calcul par itération / nombre de pixels par processeur a tendance à baisser. Ce phénomène est dû au fait que plus le VPR est important, plus les communications sont rapides puisqu'un pourcentage important de ces communications s'effectue localement à la puce. Cependant, ce phénomène se voit très peu sur la Figure 3 du fait du plus grand partage de l'unité de calcul flottant lorsque le VPR augmente.

5.3 Comparaison de la qualité des résultats

Avant de commenter les résultats obtenus sur les images, il faut souligner que :

- aucun traitement complémentaire relatif à l'affinage ou la fermeture des contours n'a été appliqué avec les algorithmes déterministes de relaxation contrairement à l'algorithme de Canny fourni par TMC (suppression des non-maxima locaux et seuillage par hystérésis).
- globalement, le GNC et le recuit par champs moyens sans interaction entre les processus de ligne donnent des résultats analogues tant en détection de contours qu'en restauration d'image.

Dans le cas d'images non bruitées (cf Annexes no 2 et 3), l'algorithme de Canny donne de bons résultats pour un temps de calcul plus faible, les seuls problèmes étant une non-fermeture des contours dans certains cas (cf Annexe 2) et une mauvaise détection lorsqu'il y a des textures ou des parcelles (cf Annexe no 3). Les algorithmes déterministes de relaxation donnent des contours plus fermés (cf Annexe no 2) mais présentent des inconvénients :

- temps de calcul plus élevé (cf tables de la section 5.2)
- pas d'affinage de contours (cf Annexe no 2)
- problèmes de détection si texture ou parcellaire (cf Annexe no 3). Cependant pour ce type d'images, il est possible de développer un autre modèle basé sur les champs de Markov et de faire de l'étiquetage.

En ce qui concerne les images bruitées (cf Annexes no 1, 4 et 5), les algorithmes déterministes de relaxation donnent de meilleurs résultats en détection de contours qu'un filtrage classique de type Canny même lorsqu'il y a un faible rapport signal sur bruit (tests effectués jusqu'à 0 dB). De plus, il n'y a pas de coût supplémentaire à payer pour obtenir une image restaurée. Il faut noter que :

- il est possible d'obtenir des résultats légèrement meilleurs avec le filtre de Canny en fixant de façon indépendante les deux seuils (ce qui n'est pas le cas de l'algorithme fourni par TMC sur la CM2). Cependant, les résultats obtenus sont moins bons qu'avec le GNC ou le recuit par champs moyens.
- pour des images difficiles telles que celles présentées dans les Annexes no 4 et 5, la qualité de la restauration avec le GNC ou le recuit par champs moyens n'est pas très bonne. Cependant, on obtient un lissage qui peut être utile notamment pour la détection de bâtiments ou de routes. En ce qui concerne la détection de contours, on observe un phénomène de saturation, typique des algorithmes de relaxation, au niveau du village (cf Annexe 4) ou de la route (cf Annexe 5).

6 Conclusion

Dans ce rapport, nous avons présenté les résultats obtenus en détection de contours et restauration d'image par deux algorithmes déterministes de relaxation basés sur une modélisation Markovienne de l'image. Les principaux avantages d'une telle approche sont:

- La prise en compte des interactions locales au niveau du pixel.
- L'utilisation des processus de ligne pour la détection des contours.
- L'incorporation de contraintes dans l'énergie.
- L'obtention d'une solution déterministe qui peut être implémentée à l'aide d'un algorithme parallèle itératif.

D'un point de vue méthodologique, nous sommes partis de la version séquentielle des algorithmes que nous avons transformée en une version parallèle. Dans les deux cas, cette transformation a été assez aisée puisqu'il ne nous a pas été nécessaire de repenser l'algorithme. Nous avons conservé sa structure en transformant chaque séquence d'opérations séquentielles sur l'ensemble des pixels par une opération parallèle équivalente. L'utilisation du langage *lisp* nous a permis d'exprimer ces algorithmes de façon plus concise et plus élégante que leur équivalent séquentiel.

Bien que l'avantage que nous ayons tiré de cette expérience soit essentiellement un gain de temps de calcul important, nous pensons que le modèle de parallélisme de données à grain fin est intéressant et apporte une nouvelle façon d'appréhender des problèmes réels. Si ce modèle est particulièrement bien adapté à la vision bas niveau, il semble prometteur également pour les problèmes de plus haut niveau où des expériences ont déjà été réalisées, comme par exemple en reconnaissance d'objets [26]. Les techniques sont alors différentes, mais la machine dispose de primitives (telles que le *"segmented scan"*) et d'une grande flexibilité de reconfiguration qui permettent d'envisager des solutions à la fois originales et *"naturelles"*.

7 Bibliographie

1. J. Besag, "Spatial interaction and the statistical analysis of lattice systems", *Jl of Royal Statistical Society* #2, pp 192-236, 1974.
2. G. Bilbro, R. Mann, T. Miller, W. Snyder, D. Van den Bout et M. White, "Optimization by mean field annealing", *Advances in Neural Information Processing Systems*, Touretzky Ed., Vol.1, pp 91-98, 1988.

3. A. Blake et A. Zisserman, "Visual reconstruction", *MIT Press, Cambridge - MA*, 1987.
4. C. Campbell, D. Sherington et K. Y. M. Wong, "Statistical mechanics and neural networks", *Neural Computing Architecture, MIT Press, Cambridge - MA*, I. Alexander Ed., pp 239-257, 1989.
5. R. Chellappa, S. Chatterjee et R. Bagdazian, "Texture Synthesis and compression using Gaussian-Markov random field models", *IEEE Trans. Systems, Man and Cybern.*, Vol SMC-15, pp 298-303, Mars/Avr. 1985.
6. D. Geiger et F. Girosi, "Parallel and deterministic algorithms for MRFs : surface reconstruction and integration", *MIT AI Memo 1114*, Juin 1989.
7. D. Geiger et F. Girosi, "Parallel and deterministic algorithms for MRFs : surface reconstruction and integration", *Proc. ECCV*, Antibes, Avr. 1990.
8. S. Geman et D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-6, pp 721-741, Nov. 1984.
9. S. Geman et C. Graffigne, "Markov random fields image models and their application to computer vision", *Proc. ICM86, Ed A.M. Gleason, Amer. Math. Soc. - Providence*, 1987.
10. P. Herault et J. Niez, "How neural networks can solve hard graph problems : a performance study on the graph K-partitioning", *Proc. Neuro-Nimes*, pp 237-255, Nîmes, Nov. 1989.
11. W.D. Hillis, "The Connection Machine ", *MIT Press, Cambridge - MA*, 1985.
12. F.C. Jeng et J.M. Woods, "Image estimation by stochastic relaxation in the compound Gaussian case", *Proc. ICASSP*, New-York, Avr. 1988.
13. R. Kindermann et J.L. Snell, "Markov random fields and their applications", *Amer. Math. Soc.*, Vol. 1, pp 1-142, 1980.
14. S. Kirkpatrick, C. Gelatt et M. Vecchi, "Optimization by simulated annealing", *Science* 220, pp 671-680, 1983.
15. J. Little, G. Belloch et T. Cass, " Algorithmic techniques for computer vision on a fine-grain parallel machine, *IEEE Trans. Pattern Analysis, Machine Intel.*, Vol. PAMI-11, pp 244-257, Mar. 1989.
16. D. Luenberger, "Linear and nonlinear programming", *Addison Wesley, 2nd Edition*, 1984.
17. J. Marroquin, S. Mitter et T. Poggio, "Probabilistic solution of ill-posed problems in computational vision", *Proc. Image Understanding Workshop, Scient. App. Int. Corp.*, Août 1985.

18. J. Marroquin, "Deterministic Bayesian estimation of Markovian random fields with applications to computational vision", *Proc. ICCV*, London, Juin 1987.
19. G. Parisi, "Statistical Field Theory", *Adisson Wesley*, 1988.
20. M. Plischke et B. Bergersen, "Equilibrium Statistical Physics", *Prentice Hall, Englewood Cliffs - NJ*, 1989.
21. T. Simchony et R. Chellappa, "Stochastic and deterministic algorithms for MAP texture segmentation", *Proc. ICASSP*, New-york, Avr. 1988.
22. T. Simchony, R. Chellappa et Z. Lichtenstein, "Pyramid implementation of optimal step conjugate search algorithms for some low level vision problems", *Proc. Conf. on Computer Vision*, Miami Beach, Dec. 1988.
23. T. Simchony, R. Chellappa et Z. Lichtenstein, "The Graduated Non Convexity algorithm for image estimation using Compound Gauss-Markov Field models", *Proc. ICASSP*, Glasgow, Mai 1989.
24. Thinking Machine Corporation, "Connection Machine, Model CM2 Technical Summary", *TMC, Cambridge - MA*, Mai 1989.
25. L.W. Tucker et G.G. Robertson, "Architecture and applications of the Connection Machine", *IEEE Computer*, pp 26-38, Août 1988.
26. H. Voorhees, D. Fritzsche et L. Tucker, "Exploiting data parallelism in Vision on the Connection Machine system", *Proc. 10th ICPR*, Atlantic City, Juin 1990.
27. A.L. Yuille, D. Geiger et H. Bulthoff, "Stereo integration, mean field theory and psychophysics", *Proc ECCV*, Antibes, Avr. 1990.
28. J. Zerubia et R. Chellappa, "Mean field approximation using Compound Gauss-Markov Random Field for edge detection and image restoration", *Proc. ICASSP*, Albuquerque, Avr. 1990.
29. J. Zerubia et R. Chellappa, "Mean field annealing for edge detection and image restoration", *Proc. EUSIPCO*, Barcelona, Sept. 1990.
30. J. Zerubia et F. Ployette, "Parallel algorithms for low level vision on a Connection Machine (CM2)", *soumis à publication à "2nd IEEE Symp. on Parallel and Distributed Processing"*, Dallas, Dec. 1990.

A Annexe 1

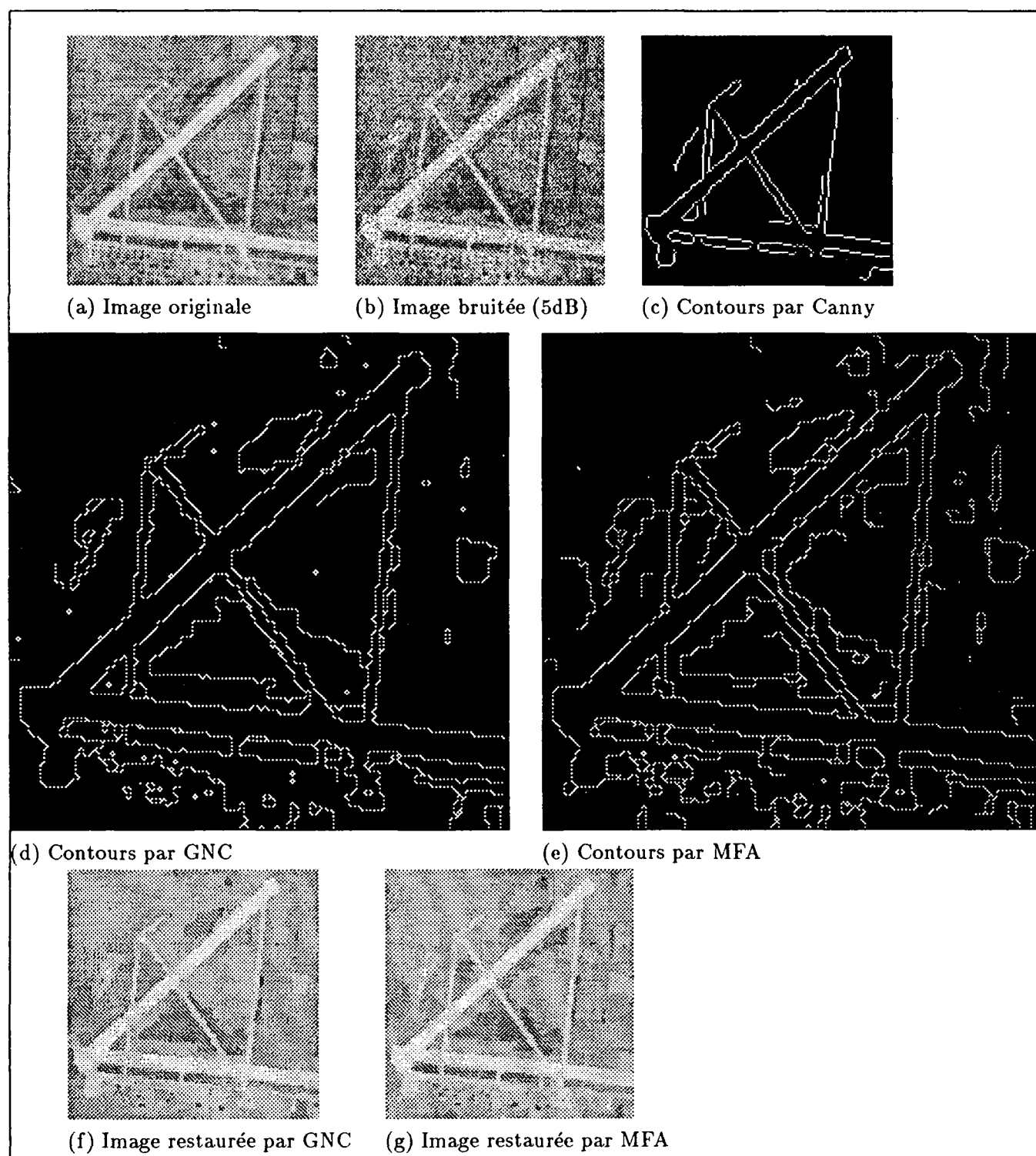


Figure 4: Image aérienne 128x128.

B Annexe 2

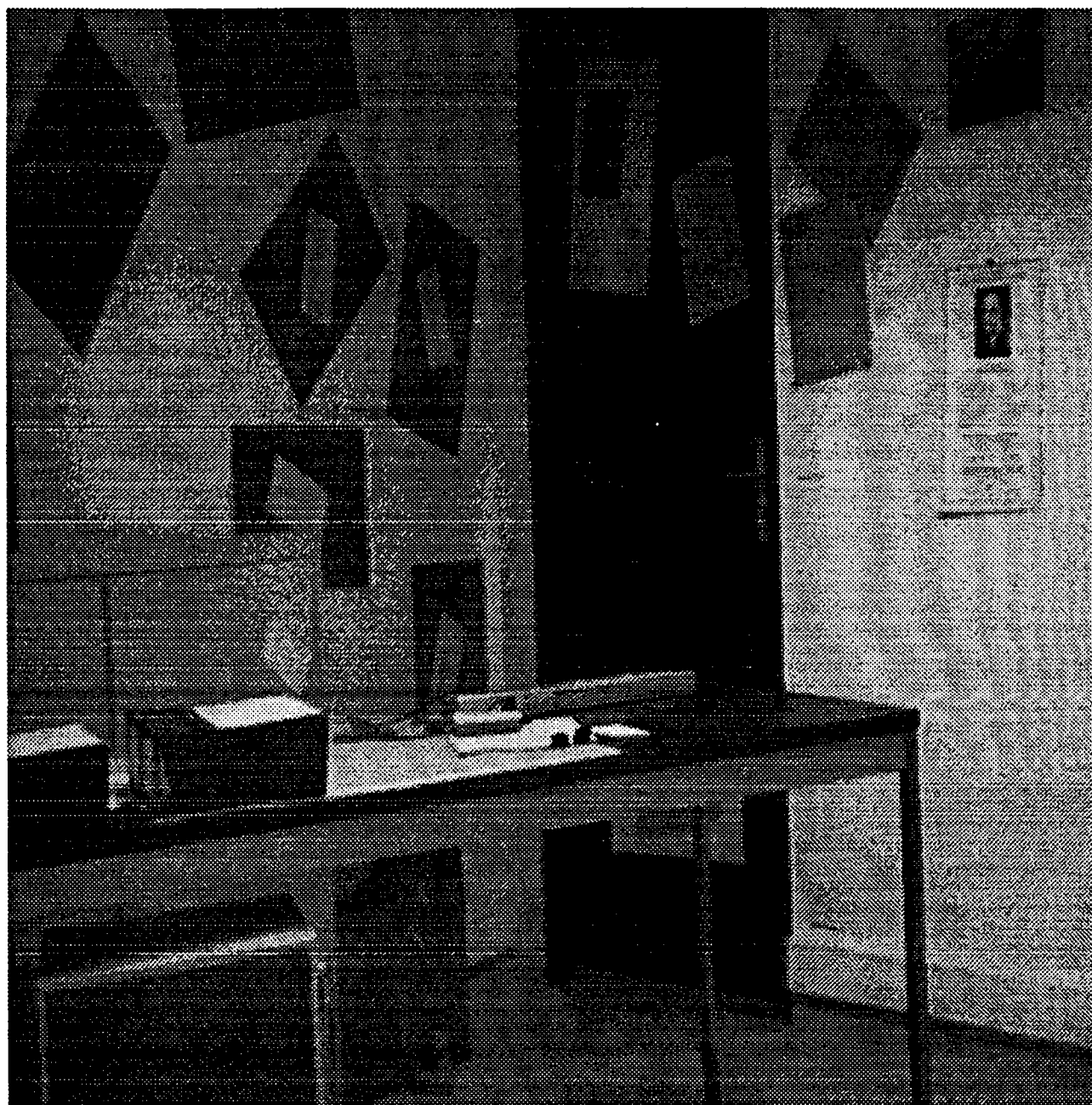


Figure 5: Image d'intérieur 512x512



Figure 6: Image d'intérieur 512x512. Contours par filtre de Canny

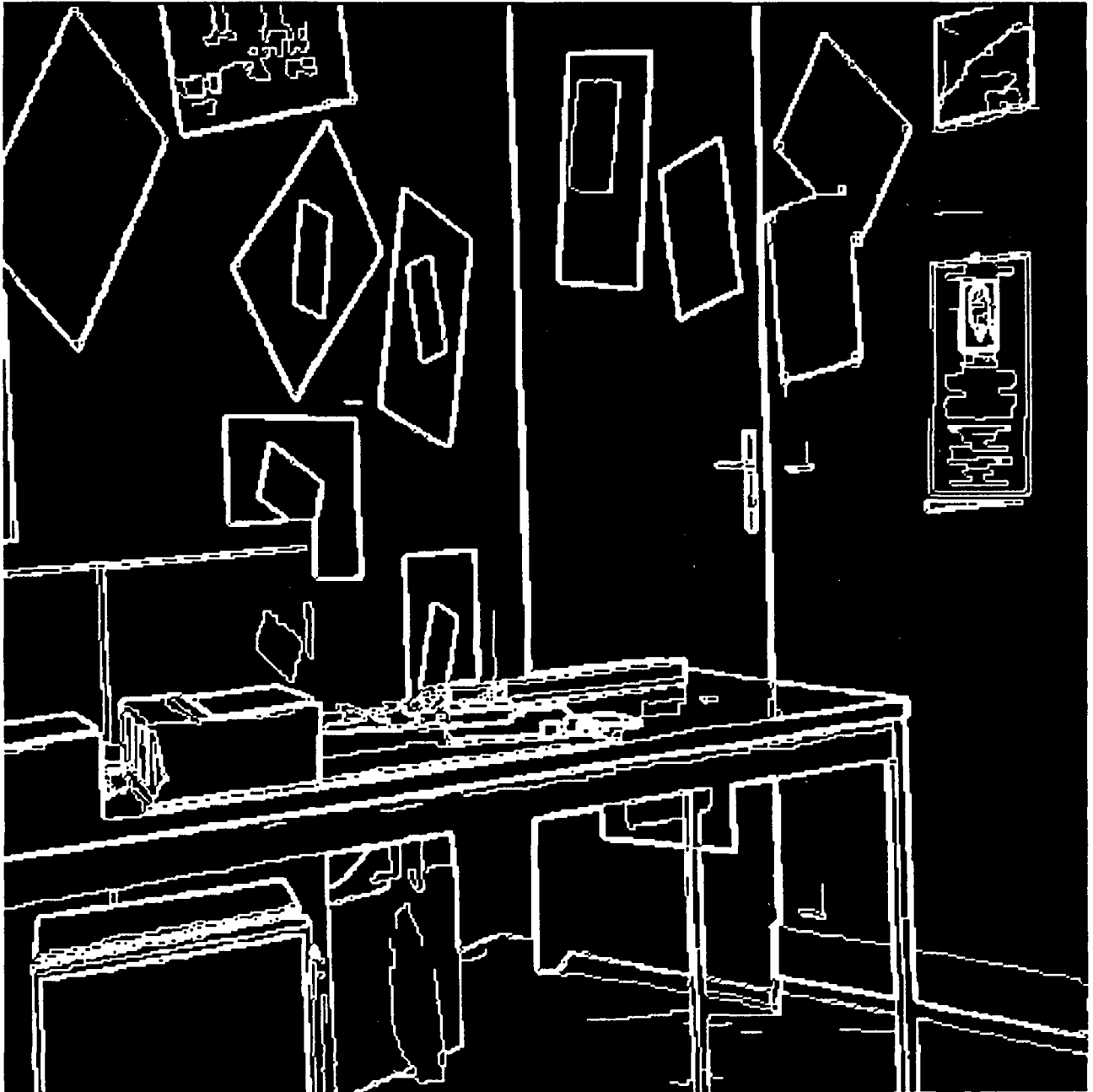


Figure 7: Image d'intérieur 512x512. Contours par GNC



Figure 8: Image d'intérieur 512x512. Contours par MFA

C Annexe 3



Figure 9: Image satellite SPOT 512x512



Figure 10: Image satellite SPOT 512x512. Contours par filtre de Canny



Figure 11: Image satellite SPOT 512x512. Contours par GNC



Figure 12: Image satellite SPOT 512x512. Image lissée par GNC



Figure 13: Image satellite SPOT 512x512. Contours par MFA



Figure 14: Image satellite SPOT 512x512. Image lissée par MFA

D Annexe 4

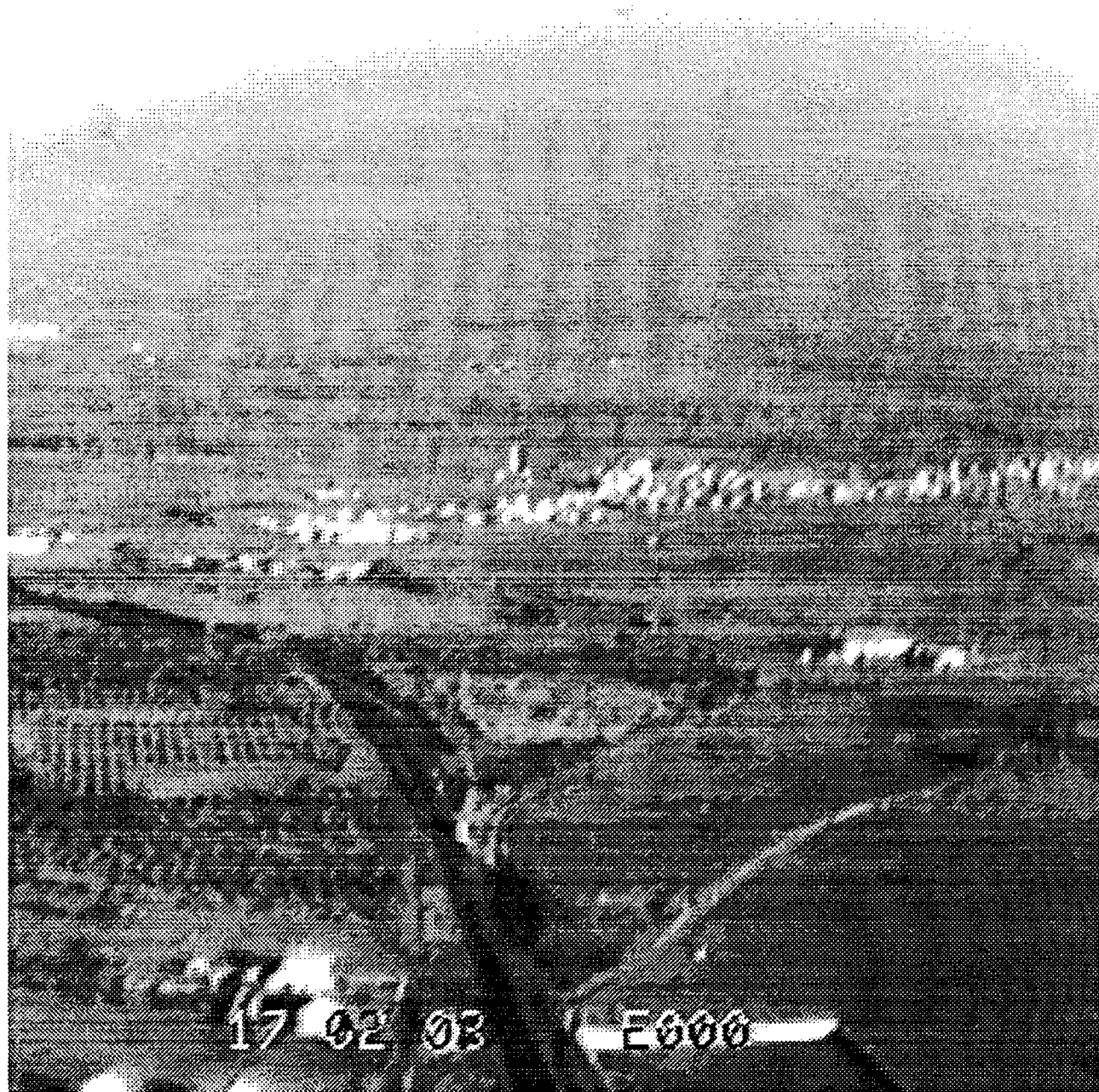


Figure 15: Image aérienne floue 512x512

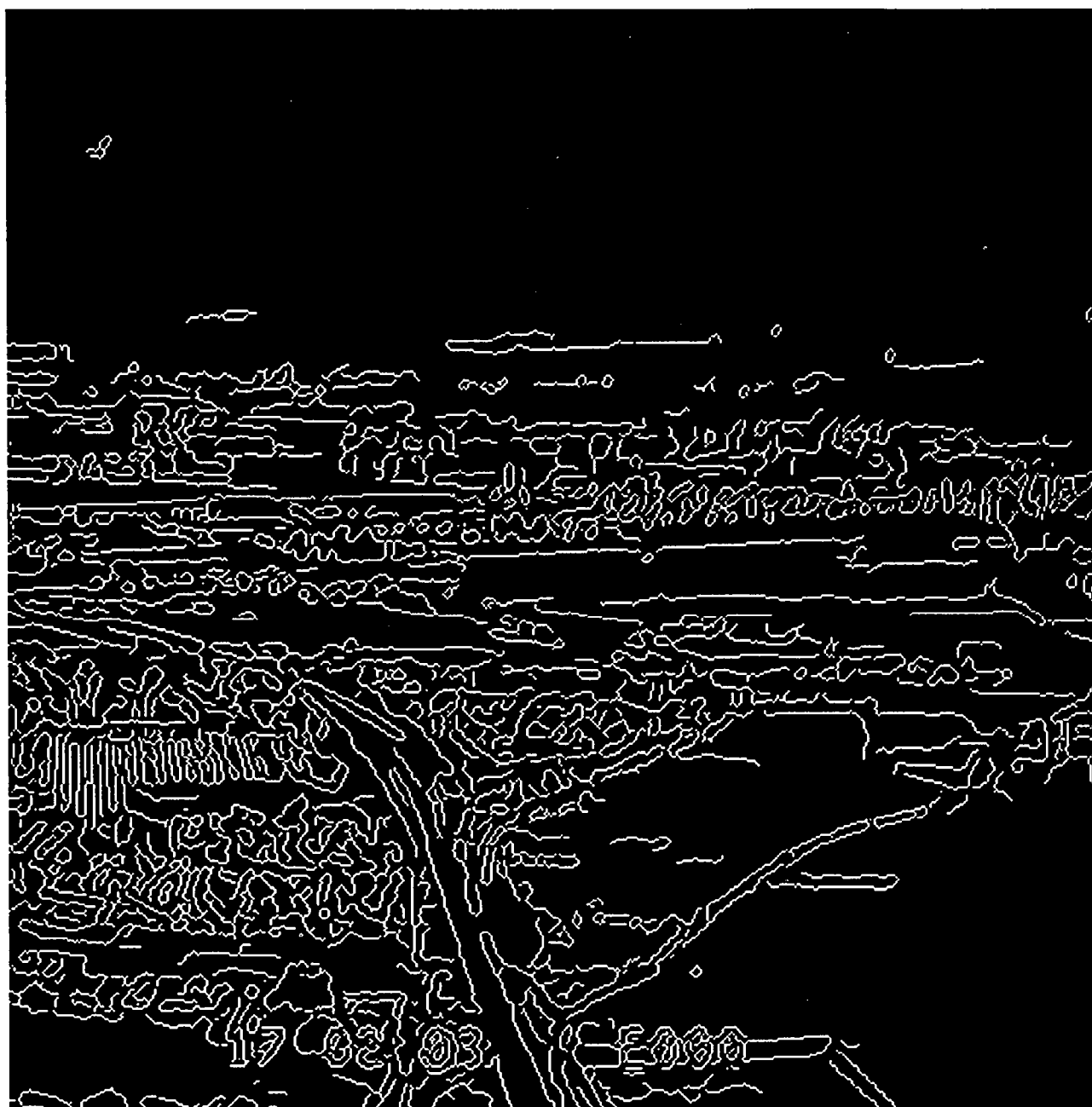


Figure 16: Image aérienne floue 512x512. Contours par filtre de Canny

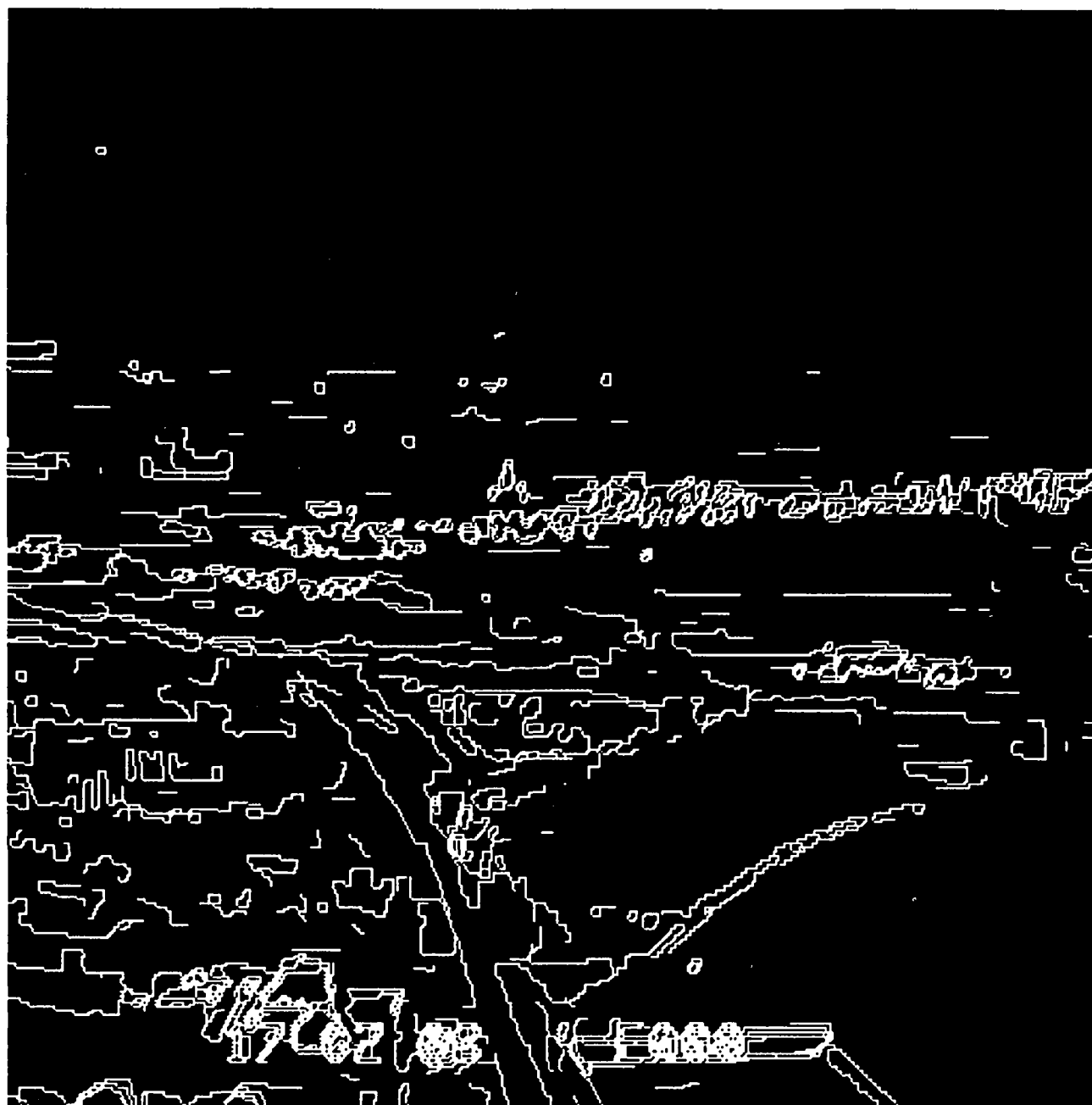


Figure 17: Image aérienne floue 512x512. Contours par GNC



Figure 18: Image aérienne floue 512x512. Image restaurée par GNC

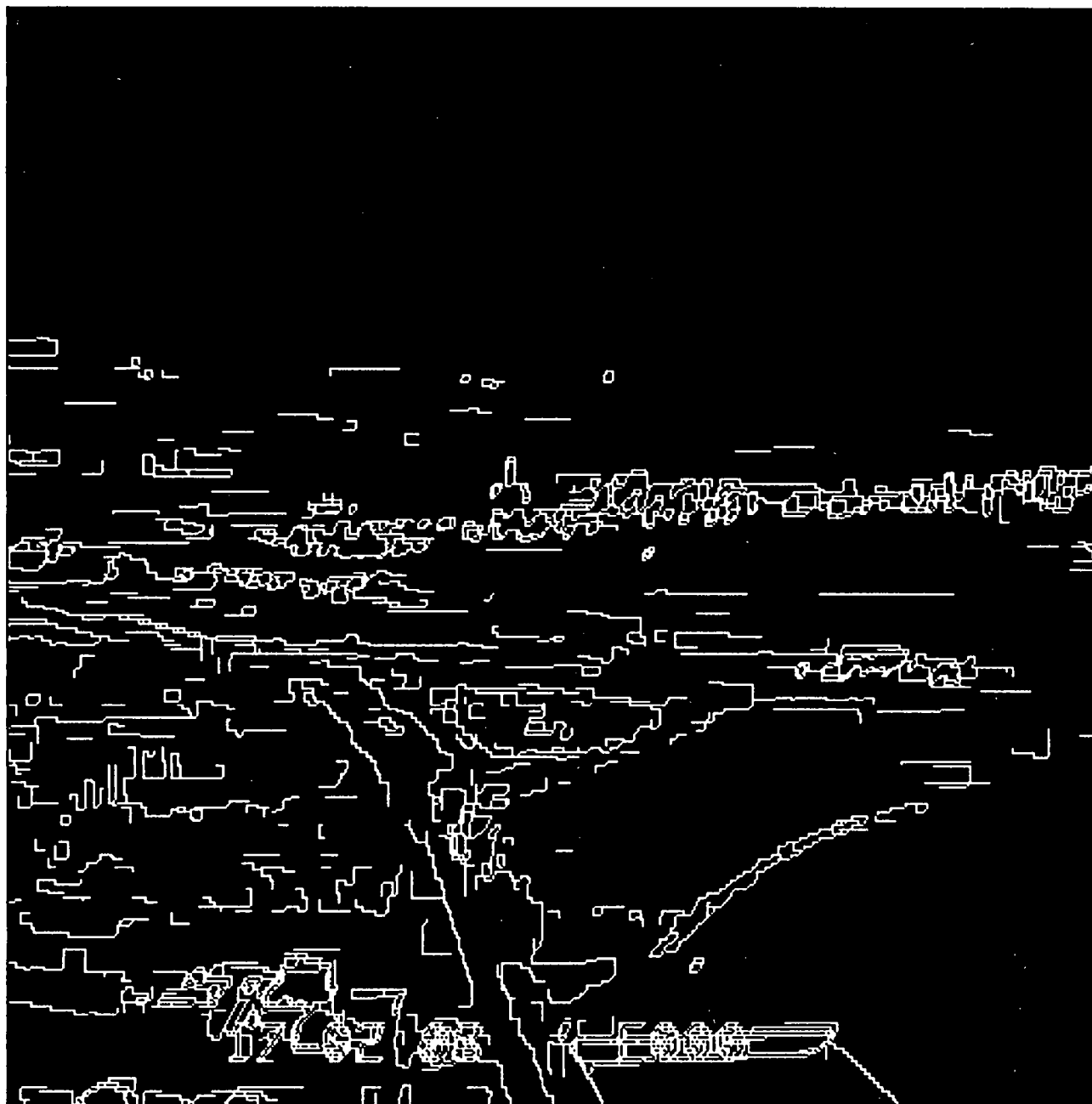


Figure 19: Image aérienne floue 512x512. Contours par MFA

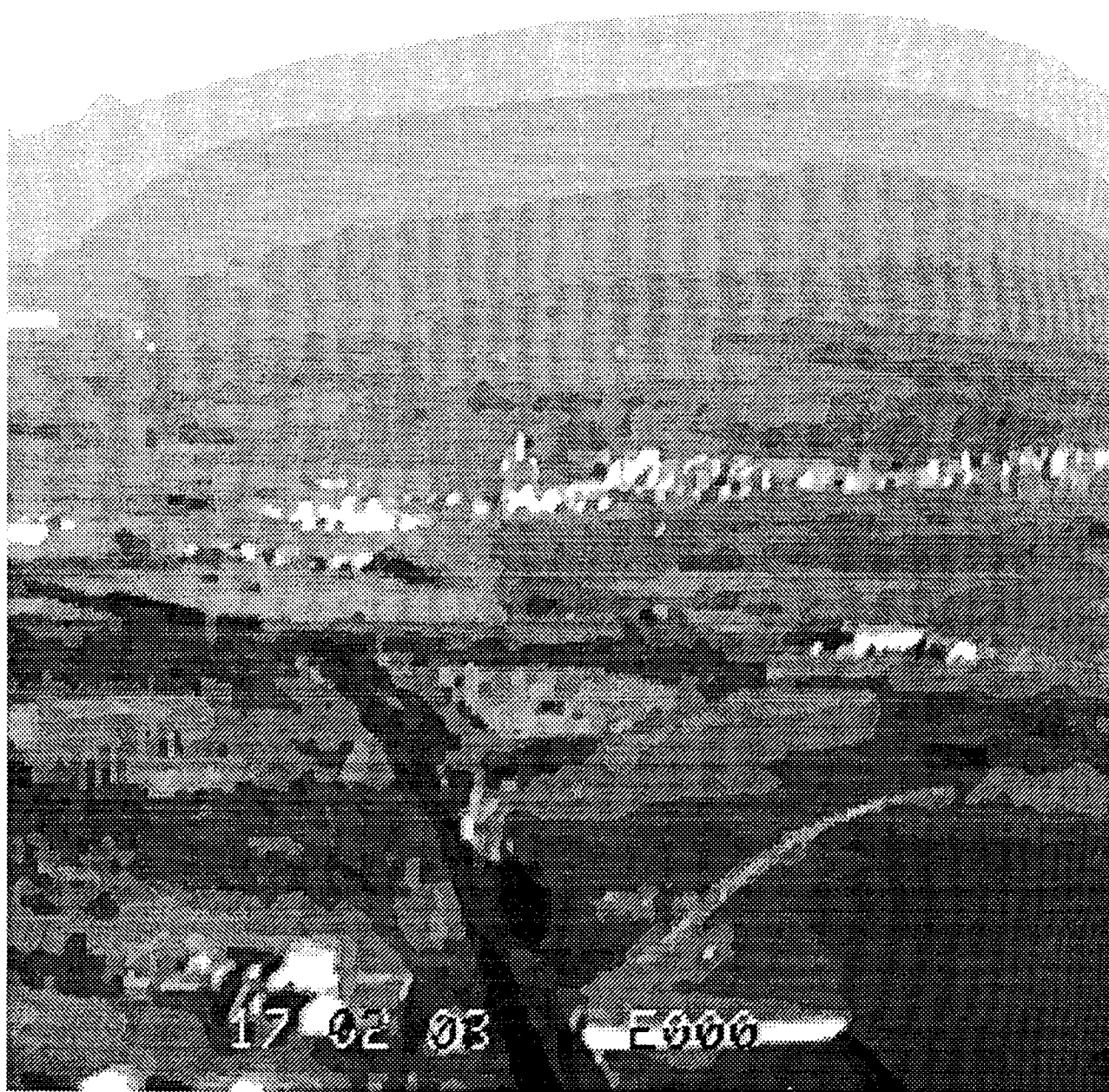


Figure 20: Image aérienne floue 512x512. Image restaurée par MFA

E Annexe 5



Figure 21: Image aérienne tramée 512x512



Figure 22: Image aérienne tramée 512x512. Contours par filtre de Canny

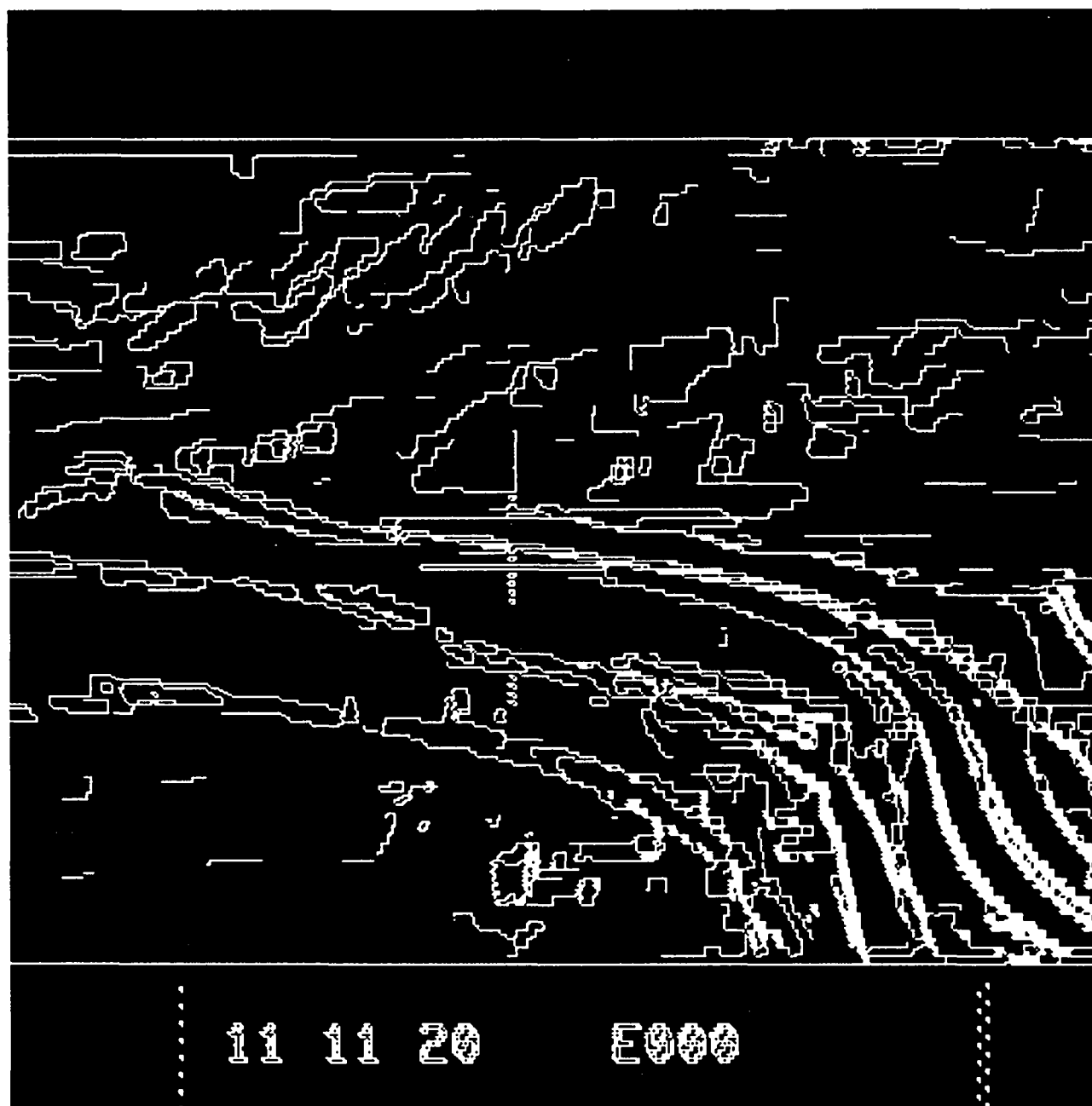


Figure 23: Image aérienne tramée 512x512. Contours par GNC



Figure 24: Image aérienne tramée 512x512. Image restaurée par GNC

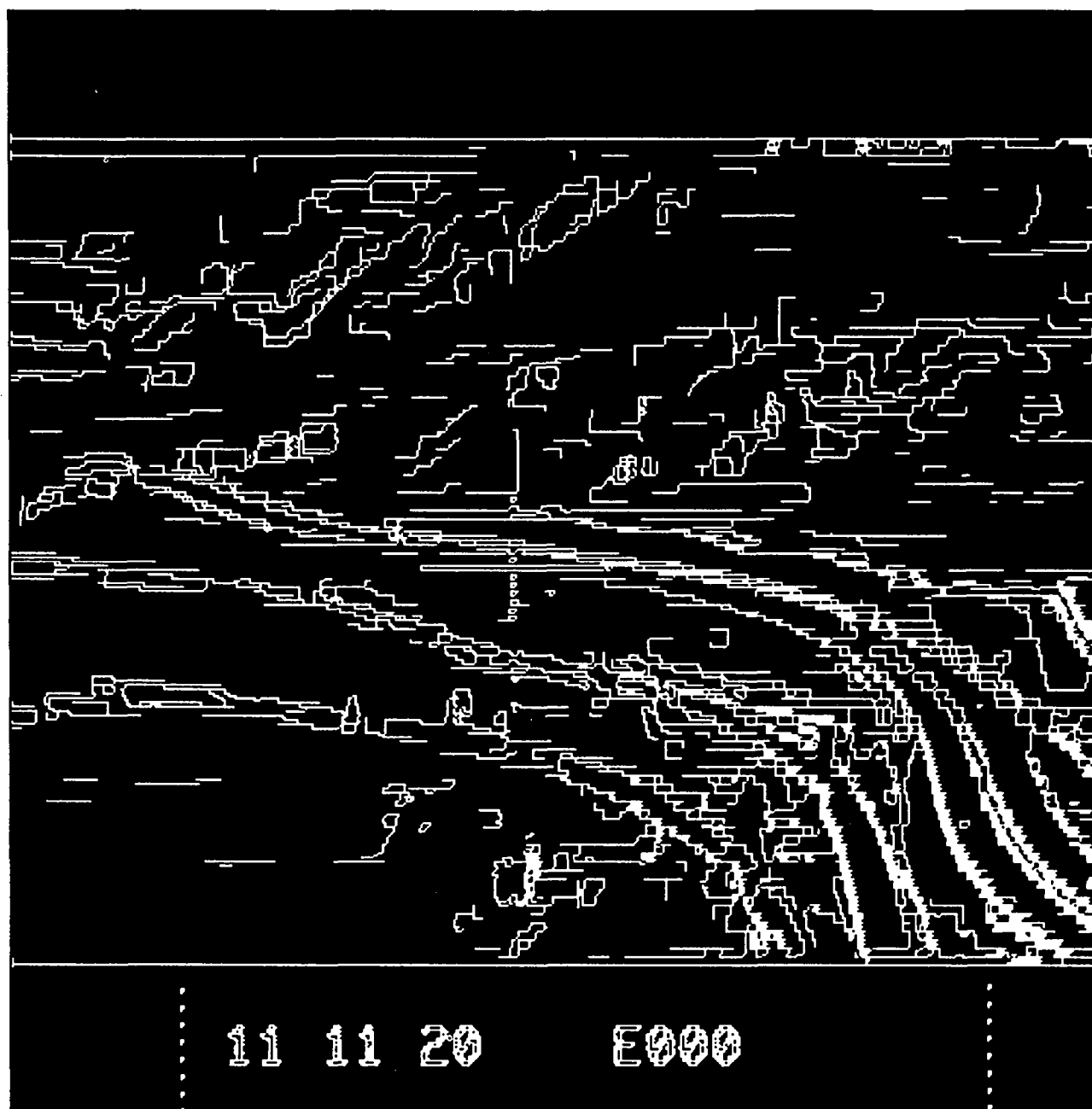


Figure 25: Image aérienne tramée 512x512. Contours par MFA

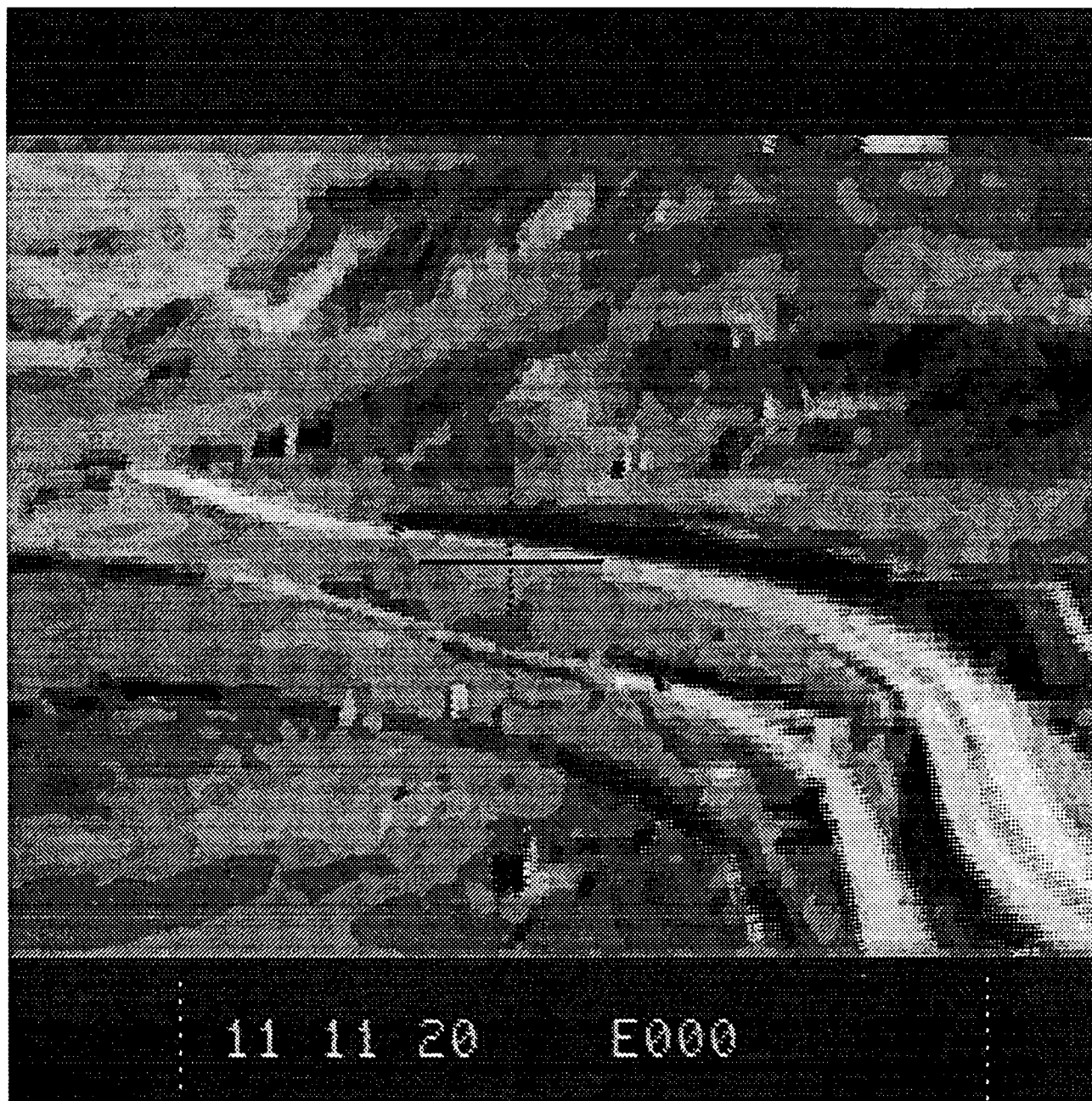


Figure 26: Image aérienne tramée 512x512. Image restaurée par MFA

ISSN 0249 - 6399